This application is submitted in the name of the following inventors:

Inventor	Citizenship	Residence City and State
McCloghrie, Keith	United Kingdom	San Jose, California
Robert, Stephan	Switzerland	Neuchatel, Switzerland
Walrand, Jean	Belgium	Berkeley, California
Bierman, Andrew		

The assignee is Cisco Technology, Inc., a California corporation having an office at 170 West Tasman Drive, San Jose CA 95134.

Title of the Invention

Sampling Packets for Network Monitoring

Background of the Invention

1. Field of the Invention

This invention relates to network monitoring.

/ / /

2. Related Art

In a computer network in which messages are transmitted and received between devices, it is often desirable to monitor the nature and volume of communication traffic. For example, by noting the number of messages (or more detailed information about those messages) transmitted from selected source devices or to selected destination devices, it can be possible to obtain useful information about usage patterns of the network. One known set of network objects used for this purpose is called RMON ("remote monitoring"). In known systems, a device coupled to and monitoring a communication link in the network generates these RMON objects. RMON objects are retrievable from the generating device using a known message protocol, such as SNMP (Simple Network Message Protocol).

RMON was originally conceived for monitoring OSI layer 1 and layer 2 communication. Accordingly, a first version of RMON (RMON1) was directed to collecting information and statistics primarily about packets between a source device MAC address and a destination device MAC address. A first version of RMON1 was optimized in some respects for Ethernet LAN communication; a second version was optimized for token-ring LAN communication. RMON1 also included capabilities for capturing the contents of selected packets, and for setting alarms upon selected events (those events being distinguished for layer 1 and layer 2 communication).

A more recent version of RMON (RMON2) extends the monitoring capabilities to include more analysis of actual packets, including identifying layer 3, layer 4, and some application aspects of communication. For example, RMON2 includes capabilities for collecting information about usage of particular routing protocols (such as IP or IPX) and particular ports used at the source device or destination device (such as ports for FTP or HTTP transactions). RMON2 also differs from RMON1 in the number of communication links that are monitored by a single device.

In parallel with the evolution from RMON1 to RMON2, another evolution has taken place: early RMON applications using RMON1 were usually directed to monitoring probes, which monitor a single port of a switch. More recent RMON applications using RMON2 are often directed to monitoring software that is embedded in a switch, and therefore is contemplated to monitor several, preferably all, interfaces of the switch.

One problem in the known art is that ability to monitor network traffic is not keeping up with the amount and speed of the network traffic itself. First, more recent versions of RMON result in an increase in the processing required for each packet. Second, it is desirable to monitor as many output interfaces as possible. Third, the bandwidth and wire speed of network interfaces is rapidly increasing due to advances in technology. All three of these effects require additional processing power in the monitoring device.

One response to this problem is to select only a sample set of packets for monitoring, rather than attempting to process all packets transmitted over the monitored communication links. The sampled traffic would serve as a proxy for all traffic, to measure the frequency of selected network events and to collect aggregate information about network traffic. United States Patent No. 5,315,580, titled "Network Monitoring Device and System", issued May 24, 1994, in the name of Peter Phaal, to assignee Hewlett-Packard Company of Palo Alto, California, shows one example of a sampling technique for monitoring.

Known sampling techniques achieve the purpose of collecting aggregate information about network traffic where the network transmission rate of packets exceeds the ability of the monitoring device to process those packets. However, these techniques suffer from several drawbacks. First, estimated frequency measurement for relatively infrequent events can be subject to error and inaccuracy. Second, processor load for the monitoring device can vary wildly in response to network traffic load. When network traffic is relatively frequent, processor load is relatively heavy, and the monitoring device can fail to keep up with the network traffic. When network traffic is relatively infrequent, processor load is relatively light, and the monitoring device can be underused.

Accordingly, it would be advantageous to provide a method and system for collecting aggregate information about network traffic, in which processor load is relatively constant despite substantial variation in network traffic, and in which the accuracy

of frequency measurement can be improved even for relatively infrequent events, due to the ability to sample more frequently. This advantage is achieved in an embodiment of the invention that samples packets from network traffic adaptively in response to that network traffic, and measures frequency in response to either the sampling rate or the frequency rate of appearance in sampled packets, or both.

Summary of the Invention

The invention provides a method and system for collecting aggregate information about network traffic, while maintaining processor load relatively constant despite substantial variation in network traffic, and capable of substantially accurate frequency measurement even for relatively infrequent events. A packet monitoring system includes an input port for receiving network packets, a sampling element for selecting a fraction of those packets for review, and a queue of selected packets. The packets in the queue are coupled to a packet-type detector for detecting packets of a selected type; the system applies a measurement technique for determining a frequency measure for those detected packets. The system includes a feedback technique for adaptively altering the sampling rate fraction, responsive to the queue length and possibly other factors, such as processor load or the detected frequency measure.

In a preferred embodiment, the measurement technique also determines an error range and a measure of confidence that the actual frequency is within the error range

of the measured frequency. The system can detect packets of multiple selected types, and provide measured frequencies and error ranges for all of the multiple selected types concurrently. Also, the measurement technique is selected so as to impose relatively little computational load per packet.

Brief Description of the Drawings

Figure 1 shows a block diagram of a system for collecting information about packet traffic.

Figure 2 shows a block diagram of a system for adaptively sampling packets.

Figure 3 shows a process flow diagram of a method for adaptively sampling packets and measuring expected frequencies for selected packet types.

Detailed Description of the Preferred Embodiment

In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. Those skilled in the art would recognize after perusal of this application that embodiments of the invention can be implemented using circuits adapted to particular process steps and data structures de-

scribed herein, and that implementation of the process steps and data structures described herein would not require undue experimentation or further invention.

Sampling System Elements

Figure 1 shows a block diagram of a system for collecting information about packet traffic.

A system 100 for collecting information about packet traffic includes a packet router or packet switch 110, a traffic management element 120, and a traffic information database 130.

The packet switch 110 includes a plurality of input interfaces 111 and output interfaces 112. The packet switch 110 it is disposed to receive a sequence of packets 113 at one or more of those input interfaces 111, and to output those packets 113 (possibly altered according to known packet rewrite rules) at one or more of those output interfaces 112. Packet routers and packet switches 110 are known in the art of computer networks.

The traffic management element 120 is coupled to at least one of the input interfaces 111 or output interfaces 112. (In a the preferred embodiment, the traffic management element 120 is coupled to substantially all of the input interfaces 111 and to sub-

stantially all of the output interfaces 112.) The traffic management element 120 is disposed to receive substantially all of the packets 113 input to the packet switch 110 and to sample a fraction of those packets 113. Similarly, the traffic management element 120 is also disposed to review substantially all of the packets 113 about to be output from the packet switch 110 and to sample a fraction of those packets 113.

In alternative preferred embodiments, the traffic management element 120 can be distributed within a plurality of devices, such that sampling of packets 113 occurs at the input interfaces 111 or output interfaces 112, while counting and analysis occur at another logical location. In such alternative preferred embodiments, the portion of the traffic management element 120 that actually samples input packets 113 marks each sampled input packet 113 as a sample and forwards those sampled input packets 113 to another portion of the traffic management element 120 for counting and analysis. Similarly, the portion of the traffic management element 120 that actually samples output packets 113 marks each sampled output packet 113 as a sample, and forwards those sampled output packets 113 back to the traffic management element 120. Sampling and forwarding of output packets 113 does not actually output a duplicate packet 113 at the output interface 112.

Since it is advantageous for the traffic management element 120 to perform accurate counting and analysis, each sampled packet 113 (whether a sampled input packet 113 or a sampled output packet 113) thus forwarded is labeled with a sequence number.

This allows the portion of the traffic management element 120 performing counting and analysis to avoid losing synchronization even if a sampled packet 113 is dropped after forwarding by the portion of the traffic management element 120 for sampling and forwarding.

The traffic management element 120 is coupled to the traffic information database 130. The traffic management element 120 is disposed to output the information it collects about sampled packets 113 to the traffic information database 130. The traffic information database 130 is disposed to store that information and to output or present that information in response to a request message 131 from a device coupled to the network (not shown).

In a preferred embodiment, the traffic information database 130 records the information about sampled packets 113 in a known format, such as the RMON MIB format, and the device coupled to the network communicates with the traffic information database 130 using a known protocol such as the SNMP protocol. The RMON MIB format and the SNMP protocol are known in the art of computer networks.

Adaptive Sampling System

Figure 2 shows a block diagram of a system for adaptively sampling packets.

A system 200 for adaptively sampling packets includes a packet input port 210, a sampling element 220, a sampled packet queue 230, an adaptive sampling controller 240, a sampled-packet output port 250, at least one packet type detector 260, and at least one frequency measure element 270.

The packet input port 210 is disposed within the traffic management element 120, and is disposed to receive substantially all of the packets 113 input to the packet switch 110. In those alternative embodiment where the traffic management element 120 is distributed in both a first portion for sampling and forwarding and a second portion for counting and analysis, the packet input port 210 is disposed within the first portion for sampling and forwarding.

In alternative embodiments, the packet input port 210 may be disposed to receive only a selected subset of the packets 113 input to the packet switch 110, such as only those packets 113 using a selected protocol such as IP or a selected protocol at another layer such as HTTP. In further or other alternative embodiments, the packet input port 210 may be disposed to receive packets 113 output by (rather than input to) the packet switch 110.

The sampling element 220 is coupled to the packet input port 210 and is disposed to sample one out of every N packets 113, where N is a control parameter. The

adaptive sampling controller 240 sets the value of N. In a preferred embodiment, the value of N is adjusted to start at a default value, and adaptively adjusted thereafter, as described herein. Thus, one out of every N packets is selected by the sampling element 220 for further processing by the traffic management element 120. In a preferred embodiment, the default value of N is selected in response to the bandwidth of the packet input port. For example, the default value can be set to 400 for a 1 gigabit-per-second port, 40 for a 100 megabit-per-second port, or 4 for a 10 megabit-per-second port.

The system 200 appends those packets 113 selected by the sampling element 220 to the tail of the sampled packet queue 230. The sampled packet queue 230 is disposed to receive, store, and present packets 113 in a FIFO (first in first out) manner. FIFO queues are known in the art of computer programming. In a preferred embodiment, the sampled packet queue 230 stores only pointers to packets 113, or pointers to packet headers, and the original packets 113 or packet headers are stored in a memory. However, the operation of the system for adaptively sampling packets is substantially similar regardless of whether the sampled packet queue 230 holds packets 113, packet headers, pointers thereto, or some related data structure.

The sampled packet queue 230 is coupled to the adaptive sampling controller 240. The adaptive sampling controller 240 compares the length of the sampled packet queue 230 against a lower threshold 231 and an upper threshold 232. The adaptive

sampling controller 240 sets the value of the control parameter N responsive to this comparison, and outputs the value of N to the sampling element 220.

In a preferred embodiment, if the length is less than the lower threshold 231, the adaptive sampling controller 240 decreases the value of the control parameter N (to sample more frequently). If the length is more than the upper threshold 232, the adaptive sampling controller 240 increases the value of the control parameter N (to sample less frequently). Methods used by the adaptive sampling controller 240 are further described with regard to figure 3. However, in alternative embodiments, the adaptive sampling controller 240 may set the value of N responsive to other factors, including any of the following (or some combination thereof):

- the actual length of the sampled packet queue 230;
- an average length of the sampled packet queue 230 for some recent time period, or some other statistical parameter for that length, such as a maximum, minimum, median, or variance thereof;
- an average number of sampled packets 113 received at the sampled packet queue 230 for some recent time period, or some other statistical parameter for that number, such as a maximum, minimum, median, or variance thereof;

- comparison of the actual or average length of the sampled packet queue 230, or the number of sampled packets 113 received at the sampled packet queue 230, with a further lower threshold (other than the lower threshold 231) or a further upper threshold (other than the upper threshold 232);

- the presence (or absence) of a packet 113 of a selected particular type (such as a special flag packet 113, a packet 113 using a known protocol such as FTP, or a multicast packet 113) received at the sampled packet queue 230, or present in the sampled packet queue 230, for some recent time period.

In a preferred embodiment, the adaptive sampling controller 240 described herein is disposed to prevent processor overloading of the traffic management element 120, by sampling at a relatively less frequent rate when packets 113 are arriving relatively more often. However, in alternative embodiments, the adaptive sampling controller 240 may be disposed for other and further purposes, such as the following:

- to obtain a more accurate count of selected particular types of packets;

- to specifically respond to expected types of network traffic (such as network traffic that is expected to be relatively bursty or relatively sparse); or

- otherwise to adapt to either the frequency or type of packets 113 seen by the traffic management element 120.

These alternative embodiments would be clear to those skilled in the art after perusing this application, would not require undue experiment or further invention, and are within the scope and spirit of the invention.

The sampled-packet output port 250 is coupled to the head of the sampled packet queue 230. The sampled-packet output port 250 couples the sampled packets 113 to one or more packet type detectors 260.

In a preferred embodiment, there is a plurality of packet type detectors 260, one for each of the selected packet types for which a frequency measurement is desired. Each packet type detector 260 counts the number of sampled packets 113 that have the selected packet type, of all those sampled packets 113 that are received. The total number of sampled packets 113 which are received is also counted, either at each packet type detector 260 or at a "universal" packet type detector 260, which counts all sampled packets 113.

Each packet type detector 260 is coupled to a corresponding frequency measure element 270, which determines an expected frequency of the selected packet type for all packets 113 in the network traffic, in response to the actual frequency of the

selected packet type for all sampled packets 113. Measurement techniques used by the frequency measure elements 270 are further described with reference to figure 3.

Figure 3 shows a process flow diagram of a method for adaptively sampling packets and measuring expected frequencies for selected packet types.

A method 300 for adaptively sampling packets and measuring expected frequencies for selected packet types includes a set of flow points and process steps as described herein. In a preferred embodiment, the traffic management element 120 (particularly the adaptive sampling controller 240 and the frequency measure elements 270) performs the method 300.

At a flow point 310, the traffic management element 120 is ready to receive a sequence (or a continuation of a sequence) of packets 113.

At a step 311, the traffic management element 120 sets the control parameter N (further described with regard to figure 2) to a preferred value of about N_0, further described below, although values of N varying substantially from N_0 are also within the scope and spirit of the invention.

At a step 312, the traffic management element 120 receives a sequence of packets 113 and samples 1 out of N of those packets 113 using the sampling element 220 to provide a stream of sampled packets 113.

At a step 313, the traffic management element 120 queues the stream of sampled packets 113 using the sampled packet queue 230, and counts the actual number of packets of each selected type using the packet type detectors 260.

At a step 314, the traffic management element 120 compares the length of the sampled packet queue 230 with the lower threshold 231 and with the upper threshold 232. In a preferred embodiment, the lower threshold 231 is constant and substantially equals a control parameter A. The traffic management element 120 performs a step 315, a step 316, or a step 317, in response to the comparison, and continues with the step 312.

If the length is less than the lower threshold 231, the traffic management element 120 performs the step 315. At the step 315, the traffic management element 120 uses the adaptive sampling controller 240 to increase the value of the control parameter N by a factor of α, where α is a control parameter. The new control parameter N is maintained for at least S new sample packets 113, where S is a control parameter.

If the length is between the lower threshold 231 and the upper threshold 232, the traffic management element 120 performs the step 316. In a preferred embodi-

ment, the upper threshold 232 is constant and substantially equals a control parameter B. At the step 316, does not adjust the control parameter N.

If the length is more than the upper threshold 232, the traffic management element 120 performs the step 317. At the step 317, the traffic management element 120 uses the adaptive sampling controller 240 to decrease the value of the control parameter N by a factor of β, where β is a control parameter. The new control parameter N is maintained for at least S new sample packets 113, where S is the control parameter described above.

In a preferred embodiment, the following values of the control parameters are used.

N_0	about 400	(as described above)
A	about 15	
α	about 2	
B	about 30	
β	about 2	
S	about 10	

The inventors have found by simulation that these values of the control parameters do not produce skew.

However, in alternative embodiments, substantially different values for these control parameters may be used; such alternative embodiments would not require undue experiment or further invention, and are within the scope and spirit of the invention.

At a flow point 320, the traffic management element 120 is ready to compute a frequency measure of packets 113 of a selected particular type.

In a preferred embodiment, the steps following the flow point 310 are performed in parallel with the steps following the flow point 320. Thus, operation of the sampling element 220 and the adaptive sampling controller 240 (to sample packets 113) is in parallel with operation of the packet type detectors 260 and their corresponding frequency measure elements 270 (to compute the frequency measure of packets 113 of each selected particular type).

At a step 321, the packet type detector 260 for a first selected type K detects a packet 113 of that type K.

At a step 322, the corresponding frequency measure element 270 for the first selected type K updates its counts of the estimated number of packets 113 of type K,

and of the actual number of total packets 113. In a preferred embodiment, the following information is maintained:

- o *count* the estimated number of packets of type K
- o *variance* the estimated variance of the count
- o *i* the number of packets
- o *j* the value of *i* for the last sampled packet of type K
- o *n* the number of sampled packets
- o *m* the value of *n* for the last sampled packet of type K

At a step 323, the frequency measure element 270 for the first selected type K determines an estimated count (from which an average frequency can be computed) for packets 113 of the selected type K, and a variance for the estimated count of packets 113 of the selected type K, according to the following sub-steps:

At a sub-step 323(a), a temporary value N_{temp} is set equal to an estimated number of packets of type K which have passed by between this sampled packet of type K and the most recent previously sampled packet of type K. In a preferred embodiment, N_{temp} is set equal to $(i - j) / (n - m)$.

At a sub-step 323(b), the estimated number of packets of type K is updated. In a preferred embodiment, *count* is set equal to *count* + N_{temp}.

At a sub-step 323(c), the estimated variance is updated. In a preferred embodiment, if $m < (n - 1)$ then *variance* is set equal to *variance* + 2 N_{temp}.

At a sub-step 323(d), the counts j and m are updated. In a preferred embodiment, m is set equal to n, and i is set equal to j.

In a preferred embodiment, the best estimate of the count is *count*, and the best estimate of the 95% confidence interval is given by *count* ± 2 **sqrt** (*variance*) where **sqrt** is a square root function.

Alternative Embodiments

Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.